Constrained Dynamics

Jungdam Won

Computer Science & Engineering

Seoul National Univ.

This material was created based on the slides and lecture notes of
Physically Based Modeling (SIGGRAPH 2001 course) by Andrew Witkin

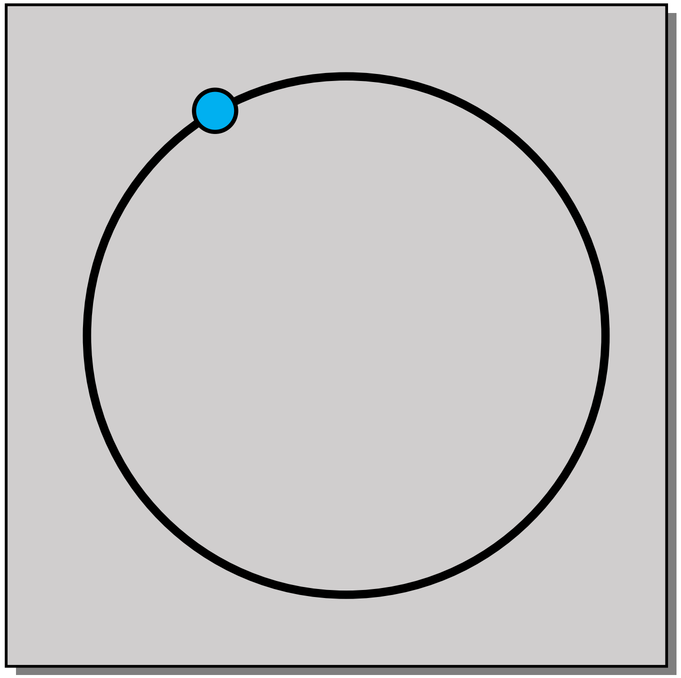
Beyond Points and Springs

- You can make just about anything out of point masses and springs, ***in principle***

- In practice, you can make anything you want as long as it's jello-like objects

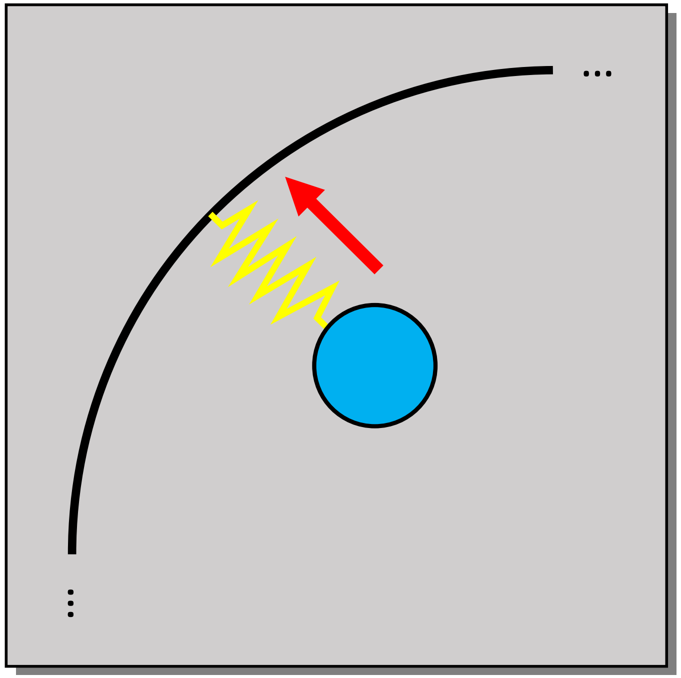
- Constraints will buy us:
  - Rigid links instead of goopy spring
  - Ways to make interesting contraptions

A Bead on a Wire



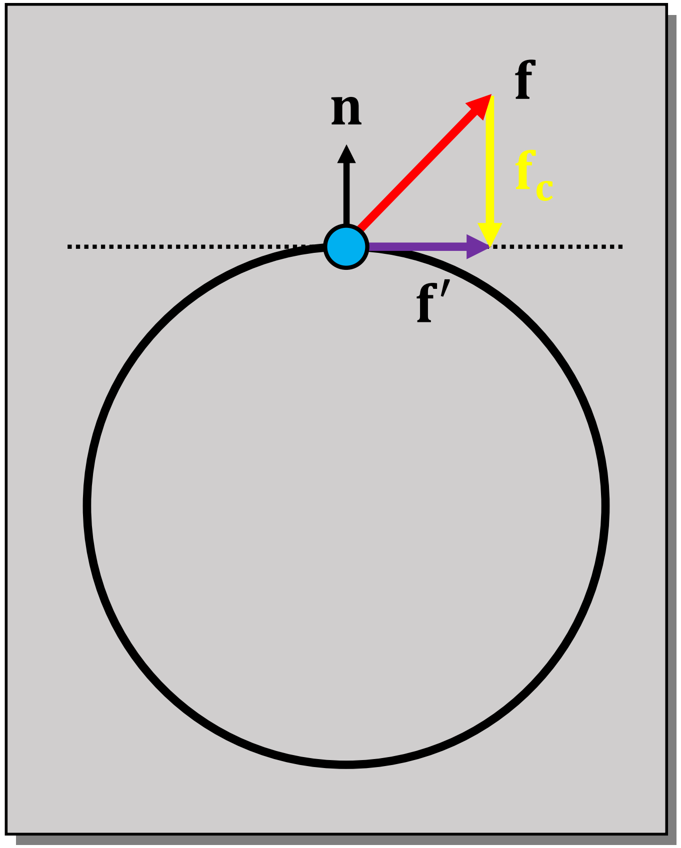
- Desired Behavior:
  - The bead can slide freely ***along*** the circle
  - It can never come off, however hard we pull
- Question:
  - How does the bead move under applied forces?

Penalty Constraints



- Why not use a spring to hold the bead on the wire?
- Problem:
  - Weak springs
    - Constraints may not be hold correctly, it should compete with other forces like gravity
  - Strong springs
    - Small timestep is required, otherwise the simulation may diverge
- A classic ***stiff system***

The Basic Trick



- Legal velocity: tangent to circle $(\mathbf{n} \cdot \mathbf{v} = 0)$
- Add a normal direction force $\mathbf{f_c}$ to project applied force $\mathbf{f}$ onto tangent: $\mathbf{f}'$
- No tug-of-war, no stiffness

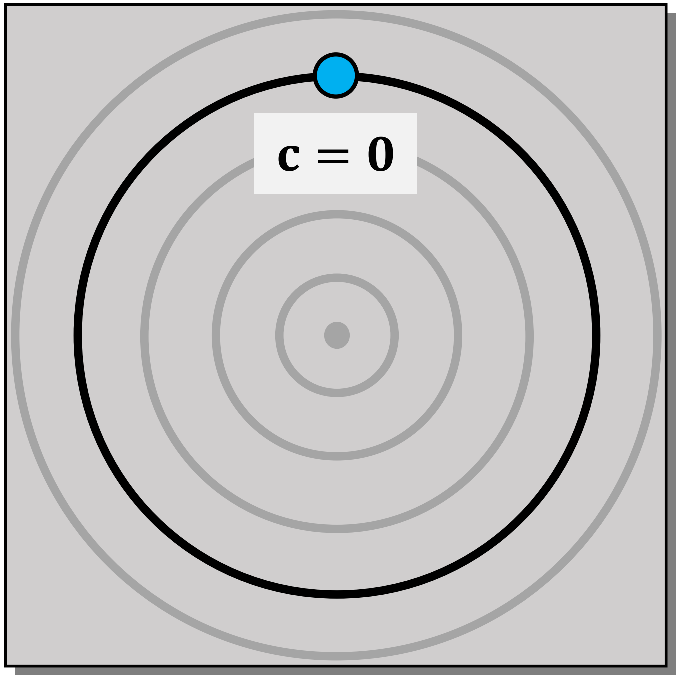
$$\mathbf{f}' = \mathbf{f} + \mathbf{f_c}$$

$$\mathbf{f_c} = -\frac{\mathbf{f} \cdot \mathbf{n}}{\mathbf{n} \cdot \mathbf{n}} \mathbf{n}$$

Now for the Algebra

- Fortunately, there's a general recipe for calculating the constraint force
- First, a single constrained particle
- Then, generalize to constrained particle systems

Representing Constraints



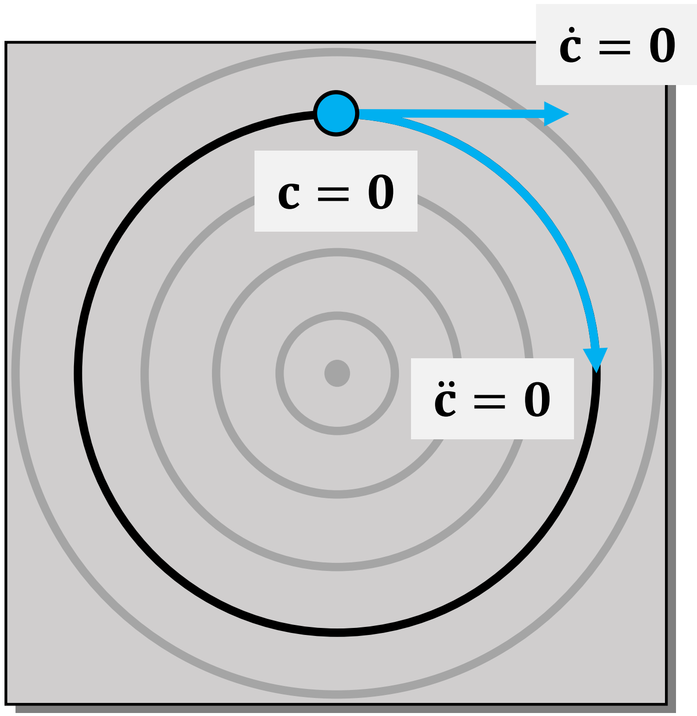
- Implicit

$$\mathbf{c}(\mathbf{x}) = \frac{1}{2}(\mathbf{x} \cdot \mathbf{x} - r^2)$$

- Parameteric

$$\mathbf{x} = r[\cos\theta \,, \sin\theta]^T$$

Energy Functions for Constraints



- Assume we start with legal position and velocity
- Setup an energy function that represents our constraints

$$\mathbf{c}(\mathbf{x}) = \frac{1}{2}(\mathbf{x} \cdot \mathbf{x} - r^2)$$

- Compute constriants forces by using the conditions below

$$\begin{aligned} \mathbf{c} &= \mathbf{0} : \text{Legal position} \\ \dot{\mathbf{c}} &= \mathbf{0} : \text{Legal velocity} \\ \ddot{\mathbf{c}} &= \mathbf{0} : \text{Legal acceleration} \end{aligned}$$

Computing Constraint Forces

$$\mathbf{c}(\mathbf{x}) = \frac{1}{2}(\mathbf{x} \cdot \mathbf{x} - r^2)$$

$$\dot{\mathbf{c}} = \mathbf{x} \cdot \dot{\mathbf{x}} = 0$$

$$\ddot{\mathbf{c}} = \dot{\mathbf{x}} \cdot \dot{\mathbf{x}} + \mathbf{x} \cdot \ddot{\mathbf{x}} = 0$$

$$\ddot{\mathbf{x}} = \frac{\mathbf{f} + \mathbf{f_c}}{m}$$

$$\ddot{\mathbf{c}} = \dot{\mathbf{x}} \cdot \dot{\mathbf{x}} + \mathbf{x} \cdot \frac{\mathbf{f} + \mathbf{f_c}}{m} = 0$$

$$\mathbf{f_c} \cdot \mathbf{x} = -m\dot{\mathbf{x}} \cdot \dot{\mathbf{x}} - \mathbf{f} \cdot \mathbf{x}$$

The number of equation is insufficient to solve unknowns!

Principle of Virutal Work

- The constraint force $\mathbf{f_c}$ never add energy to nor remove energy from the system, i.e. the constraint should be passive and lossless

$$\frac{d\mathrm{E}_k}{dt} = \frac{d}{dt}\left(\frac{m}{2}\dot{\mathbf{x}} \cdot \dot{\mathbf{x}}\right)$$

$$= m\ddot{\mathbf{x}} \cdot \dot{\mathbf{x}} = m\left(\frac{\mathbf{f} + \mathbf{f_c}}{m}\right) \cdot \dot{\mathbf{x}}$$

$$= \mathbf{f} \cdot \dot{\mathbf{x}} + \mathbf{f}_\mathrm{c} \cdot \dot{\mathbf{x}}$$

The work done by $\mathbf{f}$

The work done by $\mathbf{f}_\mathrm{c}$

$$\mathbf{f}_\mathrm{c} \cdot \dot{\mathbf{x}} = 0, \forall \dot{\mathbf{x}} \mid \mathbf{x} \cdot \dot{\mathbf{x}} = 0$$

$$\boxed{\mathbf{f}_\mathrm{c} = \lambda \mathbf{x}}$$

Computing Constraint Forces

Eq1: $\mathbf{f_c} \cdot \mathbf{x} = -m\dot{\mathbf{x}} \cdot \dot{\mathbf{x}} - \mathbf{f} \cdot \mathbf{x}$

Eq2: $\mathbf{f}_\mathrm{c} = \lambda \mathbf{x}$

$$\lambda = \frac{-m\dot{\mathbf{x}} \cdot \dot{\mathbf{x}} - \mathbf{f} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}}$$

$$\mathbf{f}_\mathrm{c} = \frac{-m\dot{\mathbf{x}} \cdot \dot{\mathbf{x}} - \mathbf{f} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}} \mathbf{x}$$

Drift and Feedback

- ***In principle***, clamping $\ddot{\mathbf{c}}$ at zero is enough
- ***In practice***, there are two problems
  - Constraints might not be met initially
  - Numerical errors can accumulate

- A feedback term handles both problems

$$\boxed{\ddot{\mathbf{c}} = -\alpha \mathbf{c} - \beta \dot{\mathbf{c}}}$$

instead of $\ddot{\mathbf{c}} = 0$

where α and β are magic numbers

Constrained Particle Systems

- Particle system: a point in state (phase) space
- Multiple constraints:
  - Each is a function $\mathbf{c}_i(\mathbf{x}_1, \mathbf{x}_2, \cdots)$
  - Legal state: $\forall_i\ \mathbf{c}_i = 0$
  - ***Simultaneous*** projection
  - Constraint force: linear combination of constraint gradients

Compact Particle System Notation

$$\mathbf{f} + \mathbf{f}_\mathrm{c} = \mathrm{M}\ddot{\mathbf{q}}$$

- $\mathbf{q}$: 3n-long ***state*** vector
- $\mathbf{f}$: 3n-long ***force*** vector
- $\mathbf{c}$: k-long ***constraint*** vector
- M: 3n x 3n ***mass*** matrix, which is diagonal

$$\mathbf{q} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}, \mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{bmatrix}, \mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_k \end{bmatrix}$$

$$\mathrm{M} = \begin{bmatrix} m_1 & & & & & & \\ & m_1 & & & & & \\ & & m_1 & & & & \\ & & & \ddots & & & \\ & & & & m_n & & \\ & & & & & m_n & \\ & & & & & & m_n \end{bmatrix}$$

Computing Constraint Forces

$$\mathbf{f} + \mathbf{f}_\mathrm{c} = \mathrm{M}\ddot{\mathbf{q}}$$

$$\dot{\mathbf{c}} = \frac{\partial \mathbf{c}}{\partial \mathbf{q}}\frac{\partial \mathbf{q}}{\partial \mathrm{t}} = \mathrm{J}\dot{\mathbf{q}} = \mathbf{0}$$

$$\ddot{\mathbf{c}} = \dot{\mathrm{J}}\dot{\mathbf{q}} + \mathrm{J}\ddot{\mathbf{q}} = \dot{\mathrm{J}}\dot{\mathbf{q}} + \mathrm{J}\mathrm{M}^{-1}(\mathbf{f} + \mathbf{f}_\mathrm{c}) = \mathbf{0}$$

$$\mathrm{J}\mathrm{M}^{-1}\mathbf{f}_\mathrm{c} = -\dot{\mathrm{J}}\dot{\mathbf{q}} - \mathrm{J}\mathrm{M}^{-1}\mathbf{f}$$

The number of equation is still not compatible with the unknowns

Principle of Virutal Work

- The constraint force $\mathbf{f_c}$ never add energy to nor remove energy from the system, i.e. the constraint should be passive and lossless

$$\frac{d\mathrm{E}_k}{dt} = \frac{d}{dt}\left(\frac{\mathrm{M}}{2}\dot{\mathbf{q}}\cdot\dot{\mathbf{q}}\right)$$

$$= \mathrm{M}\ddot{\mathbf{q}}\cdot\dot{\mathbf{q}} = \mathrm{M}\mathrm{M}^{-1}(\mathbf{f}+\mathbf{f_c})\cdot\dot{\mathbf{q}}$$

$$= \underbrace{\mathbf{f}\cdot\dot{\mathbf{q}}}_{\text{The work done by } \mathbf{f}} + \underbrace{\mathbf{f}_\mathrm{c}\cdot\dot{\mathbf{q}}}_{\text{The work done by } \mathbf{f}_\mathrm{c}}$$

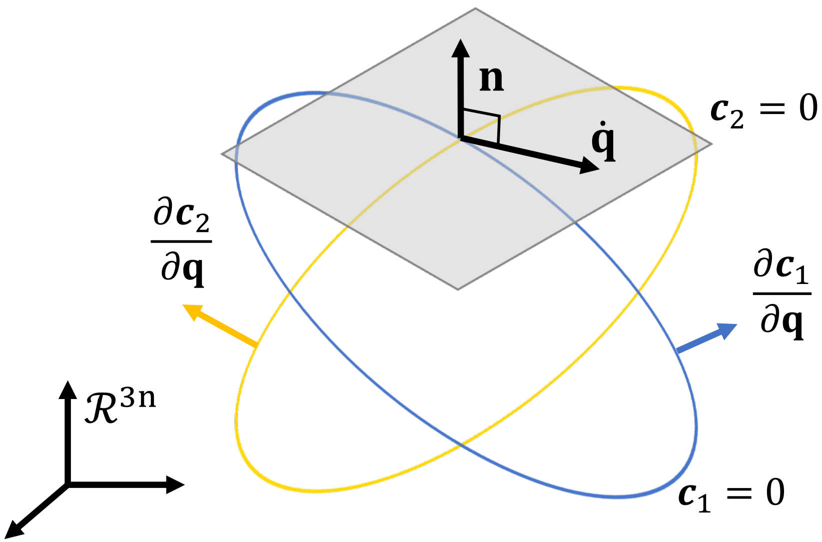
$$\boxed{\mathbf{f}_\mathrm{c}\cdot\dot{\mathbf{q}} = \mathbf{0}, \forall\dot{\mathbf{q}} \mid \mathrm{J}\dot{\mathbf{q}} = \mathbf{0}}$$

Principle of Virutal Work

$$\mathbf{f}_\mathrm{c} \cdot \dot{\mathbf{q}} = \mathbf{0}, \forall \dot{\mathbf{q}} \mid \mathrm{J}\dot{\mathbf{q}} = \mathbf{0}$$

- $\mathrm{J}\dot{\mathbf{q}}$:
  - ✓ $\dot{\mathbf{q}}$ should orthogonal to all the constraint gradients $\frac{\partial \boldsymbol{c}_1}{\partial \mathbf{q}}, \cdots, \frac{\partial \boldsymbol{c}_k}{\partial \mathbf{q}}$
  - ✓ In other words, $\dot{\mathbf{q}}$ should be a vector residing in the shared tangent space of the constraint functions
- $\mathbf{f}_\mathrm{c} \cdot \dot{\mathbf{q}}$:
  - ✓ the constraint force $\mathbf{f}_\mathrm{c}$ should orthogonal to $\dot{\mathbf{q}}$

$$\mathrm{J} = \frac{\partial \mathbf{c}}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial \boldsymbol{c}_1}{\partial \mathbf{q}} \\ \vdots \\ \frac{\partial \boldsymbol{c}_k}{\partial \mathbf{q}} \end{bmatrix}$$



$$\mathbf{f}_\mathrm{c} = \mathrm{J}^T \boldsymbol{\lambda}$$

$\mathbf{f}_\mathrm{c}$ can be made by a linear combination of the contraint graidents, and $\boldsymbol{\lambda}$ is called ***Lagarange multifliers***

Computing Constraint Forces

Eq1: $$\mathrm{J}\mathrm{M}^{-1}\mathbf{f}_{\mathrm{c}} = -\dot{\mathrm{J}}\dot{\mathbf{q}} - \mathrm{J}\mathrm{M}^{-1}\mathbf{f}$$

Eq2: $$\mathbf{f}_{\mathrm{c}} = \mathrm{J}^{T}\boldsymbol{\lambda}$$

$$\mathrm{J}\mathrm{M}^{-1}\mathrm{J}^{T}\boldsymbol{\lambda} = -\dot{\mathrm{J}}\dot{\mathbf{q}} - \mathrm{J}\mathrm{M}^{-1}\mathbf{f}$$

$$\boldsymbol{\lambda} = (\mathrm{J}\mathrm{M}^{-1}\mathrm{J}^{T})^{-1}\left(-\dot{\mathrm{J}}\dot{\mathbf{q}} - \mathrm{J}\mathrm{M}^{-1}\mathbf{f}\right)$$

$$\mathbf{f}_{\mathrm{c}} = \mathrm{J}^{T}(\mathrm{J}\mathrm{M}^{-1}\mathrm{J}^{T})^{-1}\left(-\dot{\mathrm{J}}\dot{\mathbf{q}} - \mathrm{J}\mathrm{M}^{-1}\mathbf{f}\right)$$

How can we implement all this?

Initialize all states

Until $t < T_{max}$

$\mathbf{X}(t) = \begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix}$ ← Get the current states of all particles

f ← Compute forces

$\mathrm{f_c}$ ← Compute constrint forces

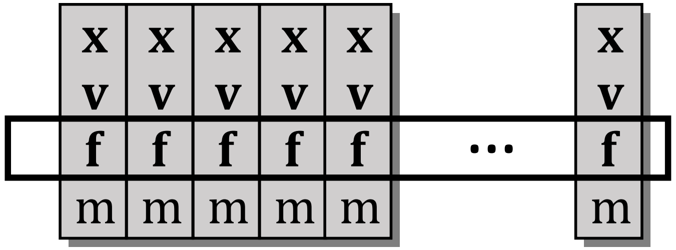
$\mathbf{X}(t+h)$ ← Solve ODEs by using $\mathbf{X}(t)$, f and other info.

Update states of all particles with $\mathbf{X}(t+h)$

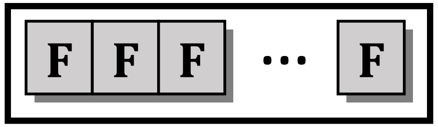
$t \leftarrow t + h$

- We also want to build models on the fly, just like masses and springs
- Approch:
  - Each constraint adds its own piece to the equation like what we created force objects

Modified Derivative Eval. Loop

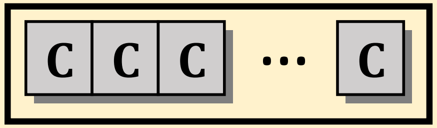


(1) Loop over particles, zero force accumulators

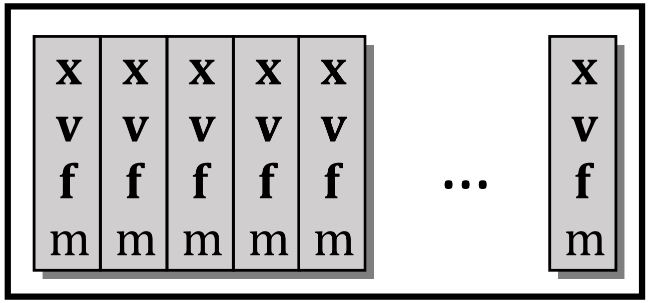


(2) Caclulate forces by invoking ***apply_force*** functions, sum all forces into accumulators

Added stuff



(3) Compute and apply constraint forces



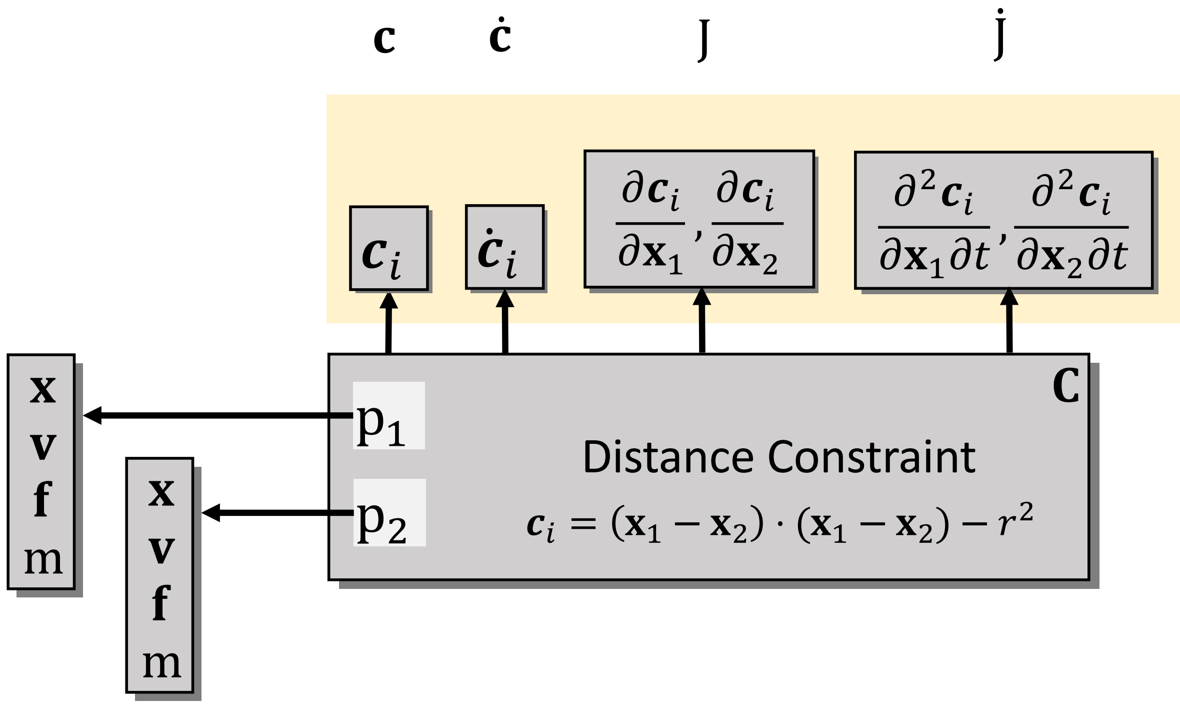
(4) Return to solver

Constraint Force Evaluation

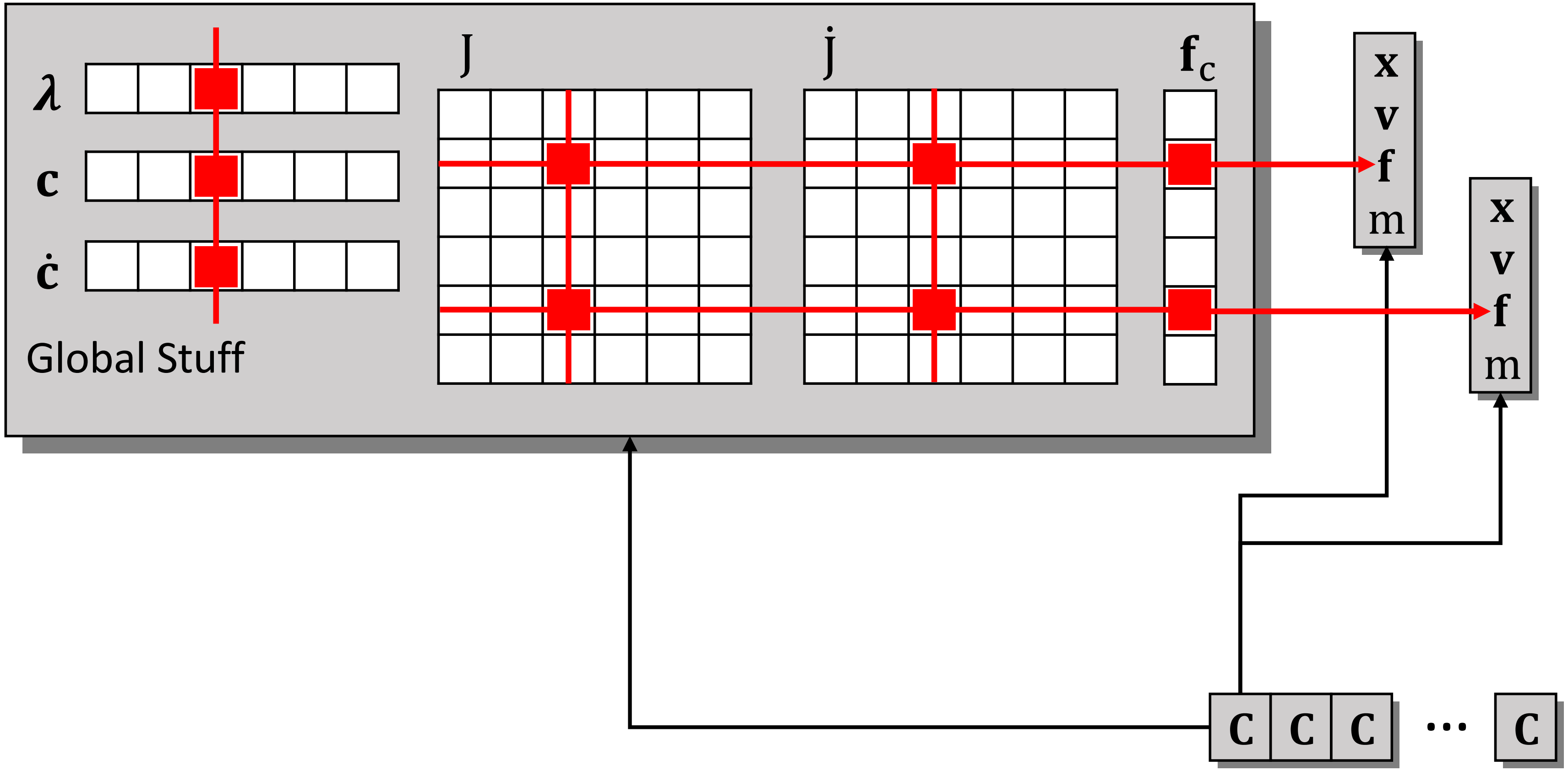
- After computing oridnary forces
  1. Loop over constrints, assemble global matrices and vectors
  2. Call matrix solver to get $\boldsymbol{\lambda}$, multiply by J^T to get constraint force
  3. Add constraint force to particle force accumulators

Constraint Structure

Each constraint must know how to compute own part

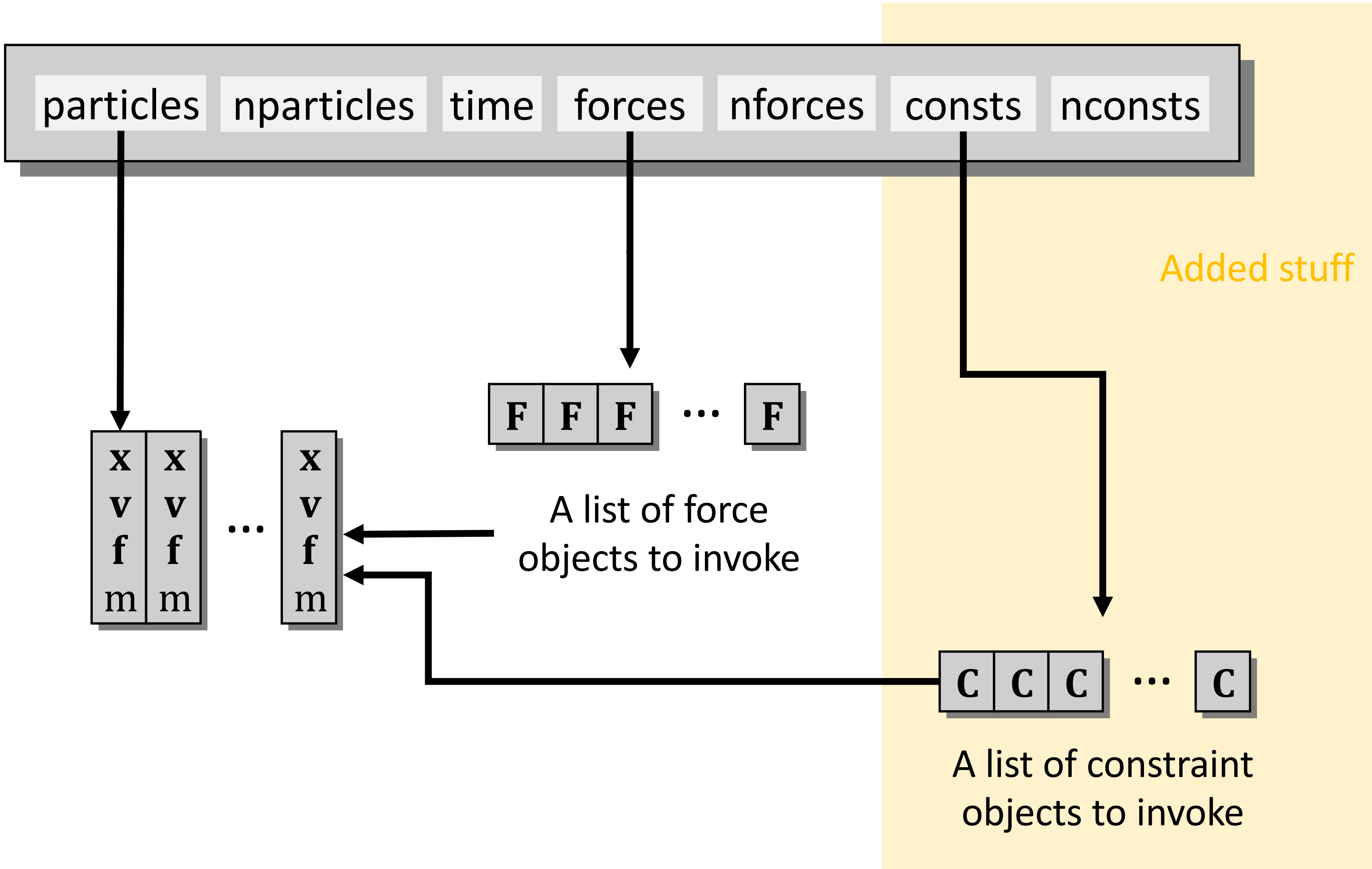


Global Structures

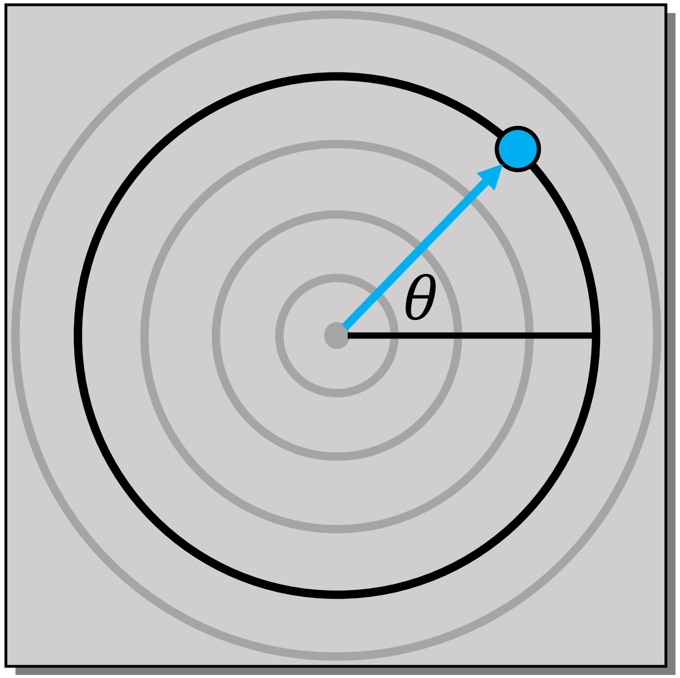


The matrices are sparse in general

Constrained Particle Systems



Representing Constraints



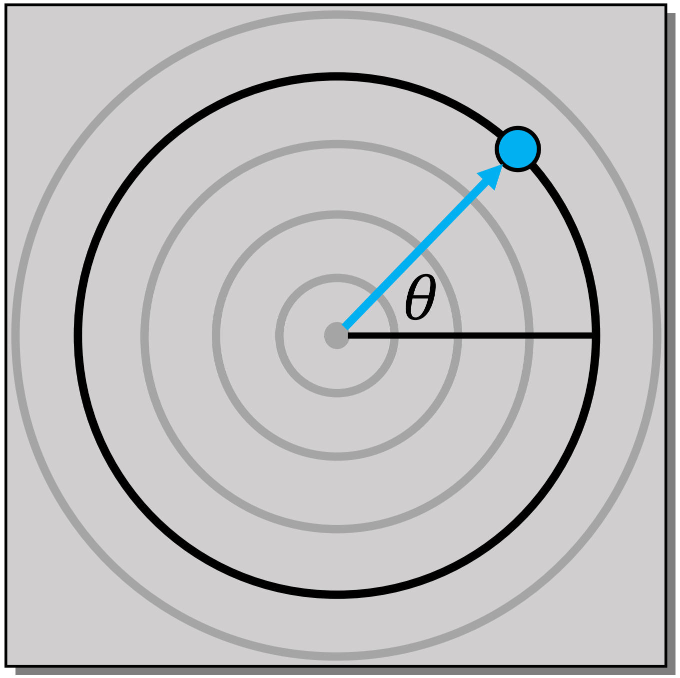
- Implicit Method

$$\mathbf{c}(\mathbf{x}) = \frac{1}{2}(\mathbf{x} \cdot \mathbf{x} - r^2)$$

- Parameteric

$$\mathbf{x} = r[\cos\theta\,, \sin\theta]^T$$

Parametric Constraints

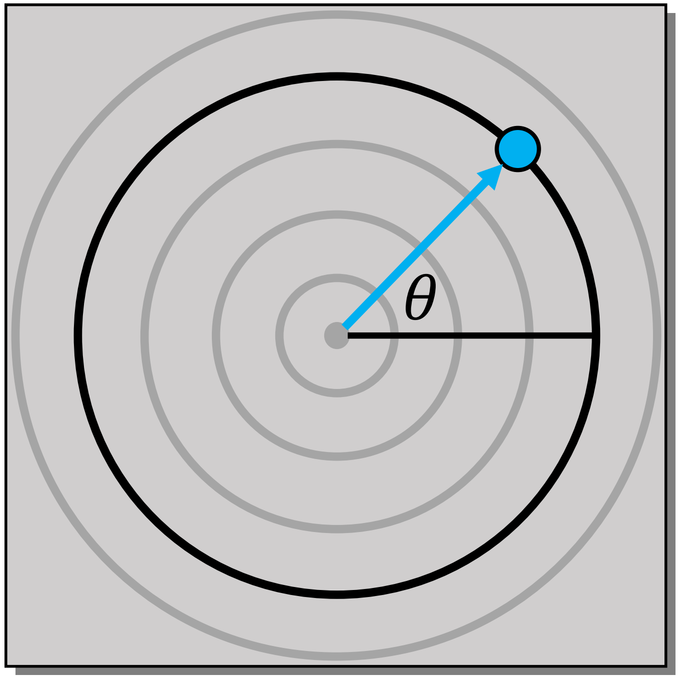


- Parametric

$$\mathbf{x} = r[\cos\theta\,, \sin\theta]^T$$

- Constraint is always met exactly
- One DoF (θ)
- Equation of motion is written by θ

Parametric bead-on-wire



- $\mathbf{x}$ is not an independent variable anymore

$$\mathbf{x} = r[\cos\theta\,, \sin\theta]^T$$

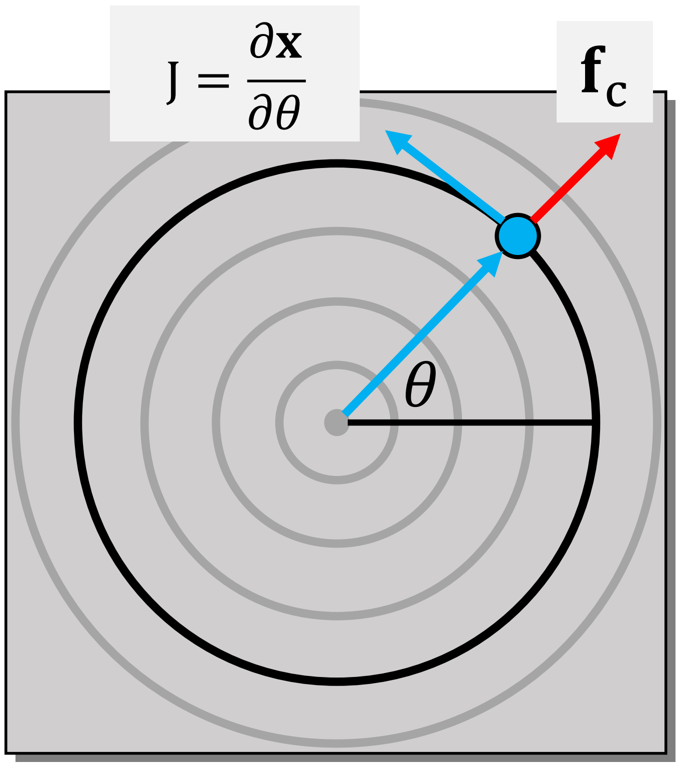
$$\ddot{\mathbf{x}} = \frac{\mathbf{f} + \mathbf{f_c}}{m}$$

$$\dot{\mathbf{x}} = \mathrm{J}\dot{\theta} = \begin{bmatrix} -r\sin\theta \\ r\cos\theta \end{bmatrix}\dot{\theta}$$

$$\ddot{\mathbf{x}} = \dot{\mathrm{J}}\dot{\theta} + \mathrm{J}\ddot{\theta}$$

$$m\mathrm{J}\ddot{\theta} + m\dot{\mathrm{J}}\dot{\theta} - \mathbf{f} + \mathbf{f_c} = 0$$

Parametric bead-on-wire



$$mJ\ddot{\theta} + m\dot{J}\dot{\theta} - \mathbf{f} + \mathbf{f_c} = 0$$

$$J^T mJ\ddot{\theta} + J^T m\dot{J}\dot{\theta} - J^T\mathbf{f} + \cancel{J^T\mathbf{f_c}} = 0$$

by the principle of virtual work

$$J^T mJ\ddot{\theta} + J^T m\dot{J}\dot{\theta} - J^T\mathbf{f} = 0$$

Solve equation for $\ddot{\theta}$

(every variables are known except for $\ddot{\theta}$)

Parametric Constraints: Summary

- Pros
  - Fewer DoF's
  - Constraints are always met
- Cons
  - Hard to formulate constriants
  - No easy way to combine constraints
- Official name: ***Lagrangian Dynamics***